AN 370: Using the Serial FlashLoader with the Quartus II Software

© April 2009 AN-370-3.1

Introduction

Using the JTAG interface, the Altera® Serial FlashLoader (SFL) is the first in-system programming solution for Altera serial configuration devices. The SFL solution is available with the Quartus® II software version 4.1 SP1 and later. The SFL megafunction is available with the Quartus II software version 6.0 and later.

Because serial configuration devices do not support the JTAG interface, the conventional method to program them is via the active serial (AS) programming interface. With the AS programming interface, the configuration data used to program serial configuration devices is downloaded via programming hardware.

However, with the SFL you can program serial configuration devices in-system via the JTAG interface. To do so, use an FPGA as a bridge between the JTAG interface and the serial configuration device.

Table 1 lists the advantages and disadvantages of both methods.

Table 1. Advantages and Disadvantages

Method	Advantage	Disadvantage
Conventional: (AS Programming)	Simple and fast	Requires separate programming interface to configure FPGAs and program serial configuration devices.
SFL solution: (JTAG Programming)	Able to configure the FPGA and program serial configuration devices using the same JTAG interface	Slow, because the SFL solution must configure the FPGA before programming serial configuration devices.

In version 9.0 and onwards of the Quartus II software, the enhanced mode of the SFL solution is introduced. This allows faster EPCS programming time with the following advantages:

- Enhanced SFL solution correctly interprets extra padding bits introduced by third programmer tool to ensure successful EPCS programming with SFL solution.
- Enhanced SFL allows conversion from JTAG Indirect Configuration (**.jic**) to Jam™ STAPL (**.jam**), JAM Byte-Code File (**.jbc**) or Serial Vector Format File (**.svf**) for multiple devices in JTAG chain in which only one device uses the SFL solution.
- Enhanced SFL allows conversion from **.jic** to **.jam**, **.jbc**, or **.svf** file for multiple devices in JTAG chain in which two or more devices uses the SFL solution.

For more information on how to enable the enhanced SFL mode, refer to "Using the SFL Megafunction in the Quartus II Software" on page 7 and "Programming Serial Configuration Devices with the Quartus II Programmer" on page 14.

The SFL supports FPGA families that configure using active serial configuration scheme. With the SFL megafunction, you can instantiate SFL image into user design. This feature allows you to perform SFL programming without resetting your design in the FPGA. The SFL solution provides more hardware programming options. For example, you can use the ByteBlaster™ II or USB-Blaster™ download cable, production tester, and other tools that have a JTAG interface.

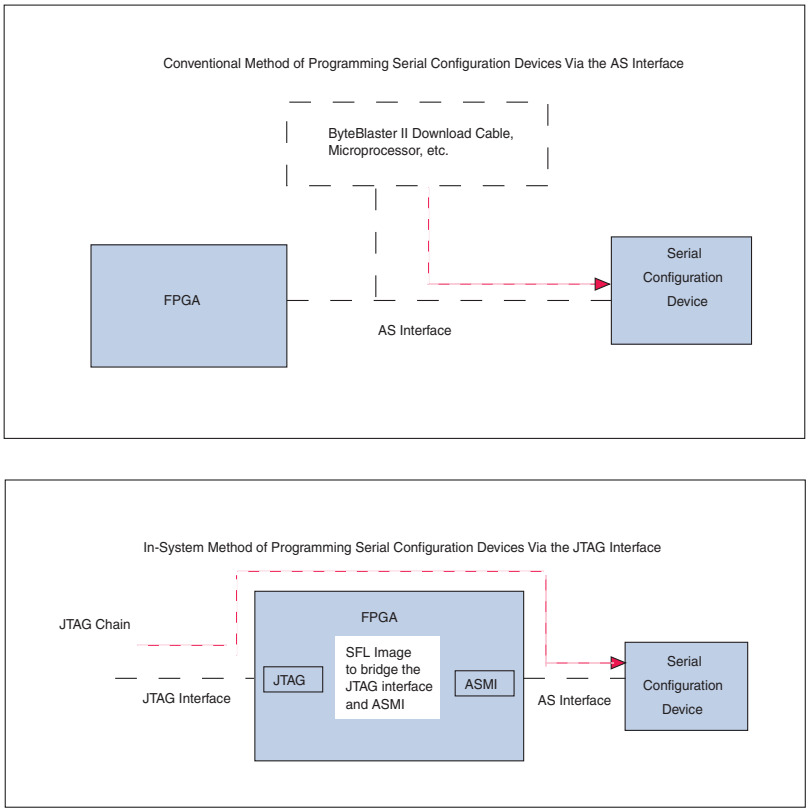
Whenever the term "serial configuration device or devices" is used in this document, it refers to Altera EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128 devices.

This application note discusses the following topics:

- "Programming Single and Multiple Serial Configuration Devices with the SFL Solution" on page 3
- "Using the SFL Megafunction in the Quartus II Software" on page 7
- "Generating .jic and .jam Programming Files in the Quartus II Software" on page 10
- "Programming Serial Configuration Devices with the Quartus II Programmer" on page 14

Figure 1 shows both the conventional method of programming serial configuration devices as well as the in-system programming method using the SFL solution.

Figure 1. Conventional Versus the In-System Programming Method



Programming Single and Multiple Serial Configuration Devices with the SFL Solution

This section describes the three steps to program both single and multiple serial configuration devices with the SFL solution.

To use the SFL solution, ensure that your board setup is in AS mode.

To program serial configuration devices using the SFL solution, perform the following steps (refer to Figure 2, Figure 3, and Figure 4):

1. To bridge the JTAG interface with the active serial memory interface (ASMI) block in the FPGA device, configure the SFL image into the FPGA. The previous design is replaced with the SFL image.
2. Program the serial configuration device or devices via the SFL image's JTAG-ASMI bridge.

   You can bypass this step if the SFL image exists in the FPGA.

3. Reconfigure the FPGA with the new configuration data. This replaces the SFL image with the new design. To reconfigure the FPGA with the new configuration data, pull the `nConfig` pin low and release it to start configuration.

Figure 2 shows the SFL programming flow.

Figure 2. Serial Flash Loader Programming Flow

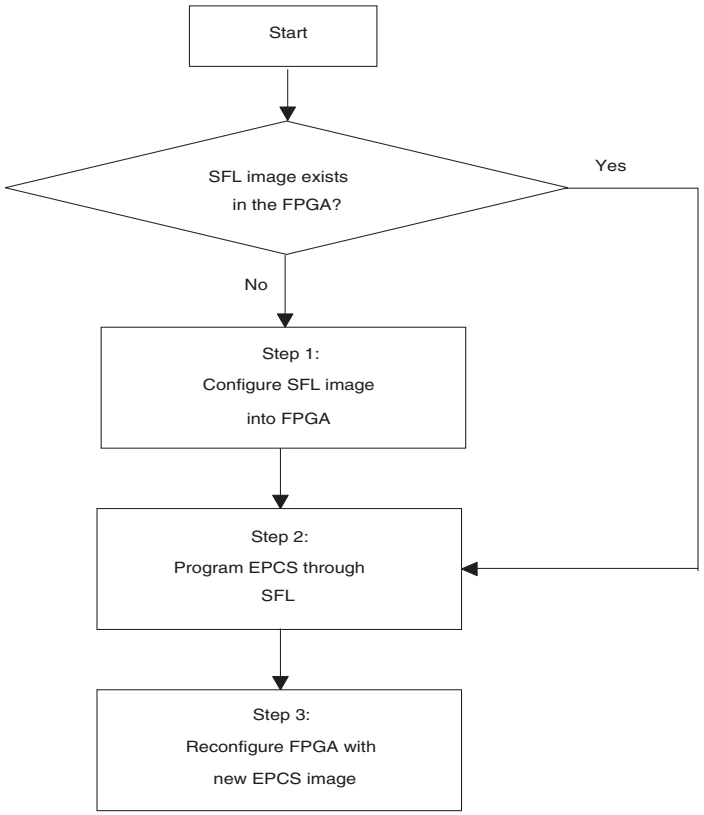
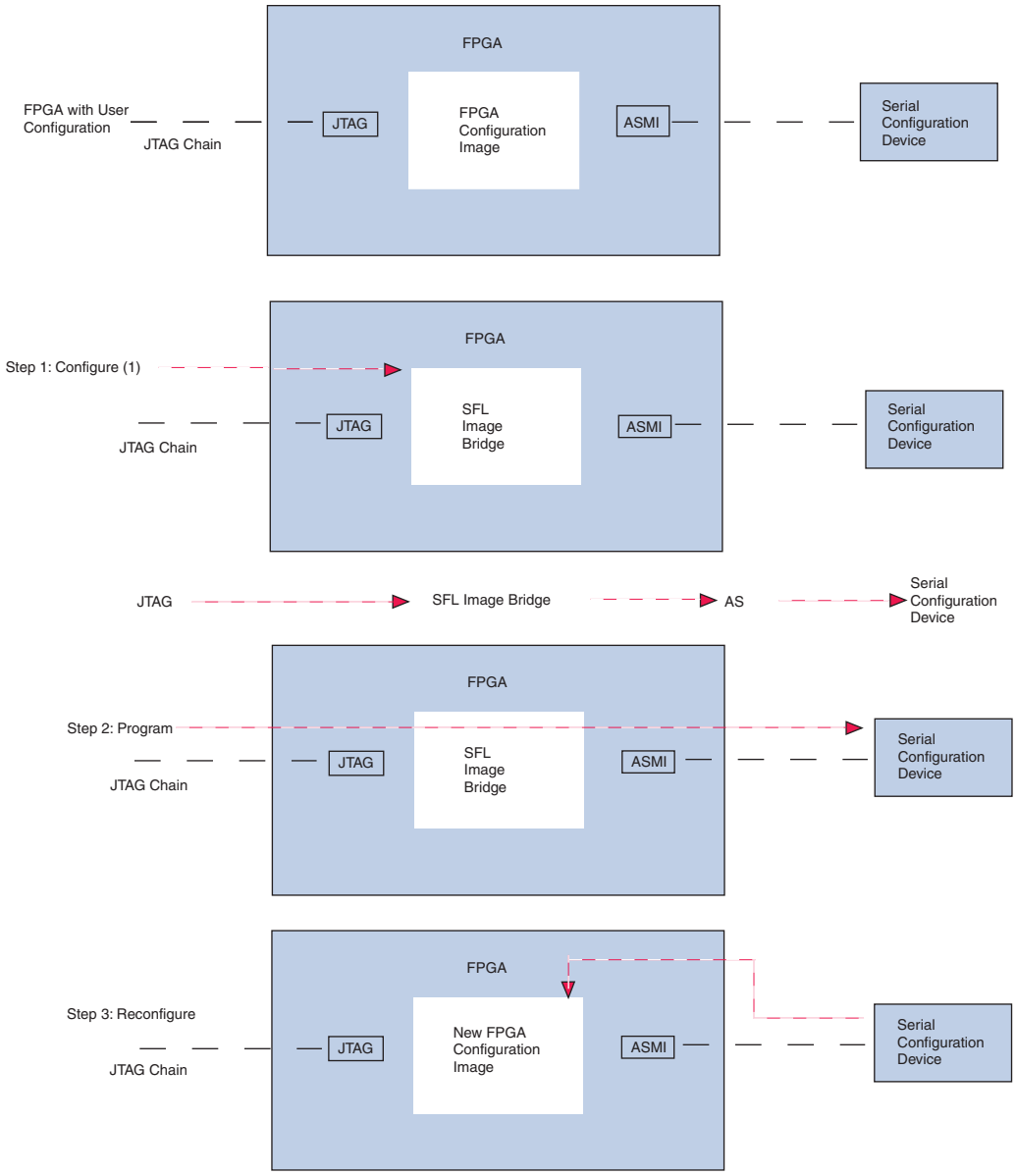

Start
SFL image exists in the FPGA?
Yes
No
Step 1: Configure SFL image into FPGA
Step 2: Program EPCS through SFL
Step 3: Reconfigure FPGA with new EPCS image

Figure 3 shows the programming of a single serial configuration device with the SFL solution.

Figure 3. Programming a Single Serial Configuration Device with the SFL Solution

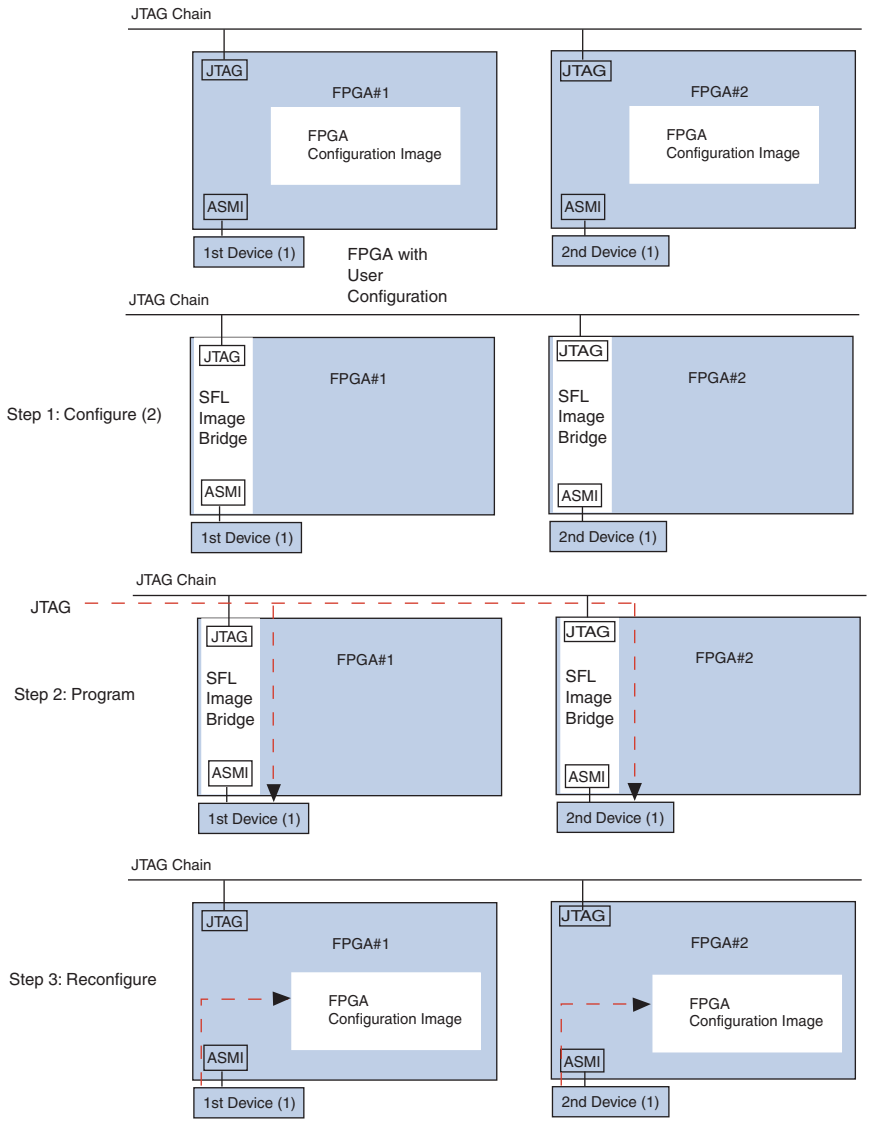


Note to Figure 3:

(1) You can bypass this step if the SFL image exists in the FPGA.

Figure 4 shows the process for programming multiple serial configuration devices with the SFL solution.

Figure 4. Programming Multiple Serial Configuration Devices with the SFL Solution



Notes to Figure 4:

(1) "1st device" and "2nd device" represent serial configuration devices.

(2) You can bypass this step if the SFL image exists in the FPGA.

Using the SFL Megafunction in the Quartus II Software

The SFL megafunction allows you to instantiate the SFL image into your design. This feature allows SFL programming without resetting your design in the FPGA with the SFL image.

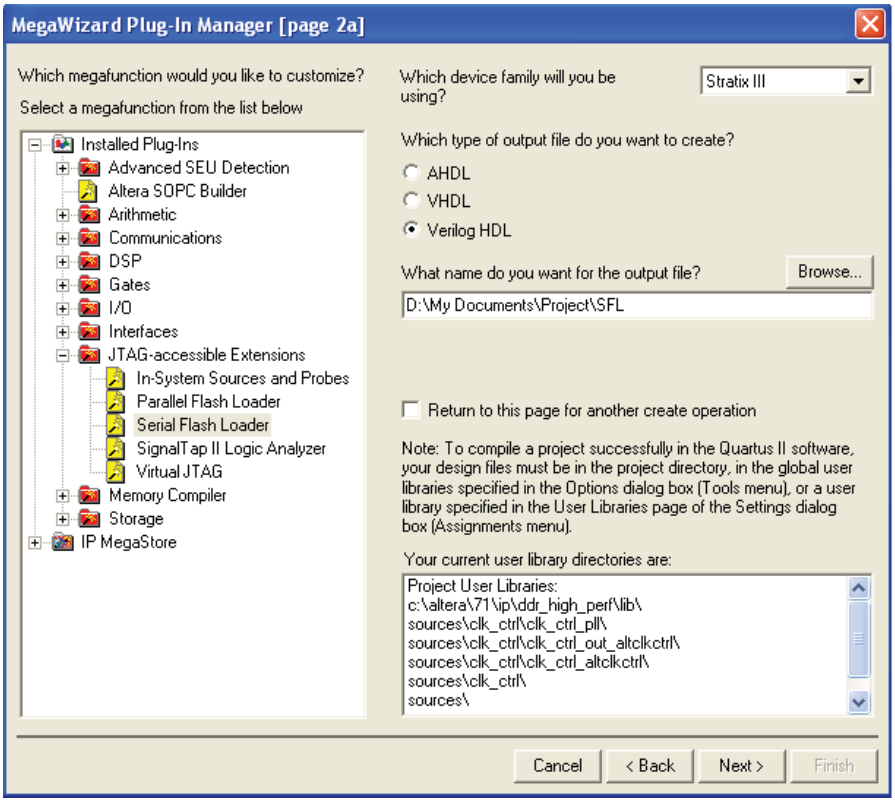
☞ The SFL megafunction must be instantiated into your design only when you must perform the EPCS re-programming without interrupting your design running in user mode. If resetting the FPGA is not an issue for you during the EPCS programming or re-programming using **.jic** or **.jam** file, you do not need to instantiate the SFL into your design. For more information on how to generate the **.jic** or **.jam** file, refer to the "Generating .jic and .jam Programming Files in the Quartus II Software" on page 10.

Instantiating SFL Megafunction in the Quartus II Software

Perform the following steps to generate an SFL megafunction instantiation. You must then instantiate the SFL megafunction in your FPGA top-level design.

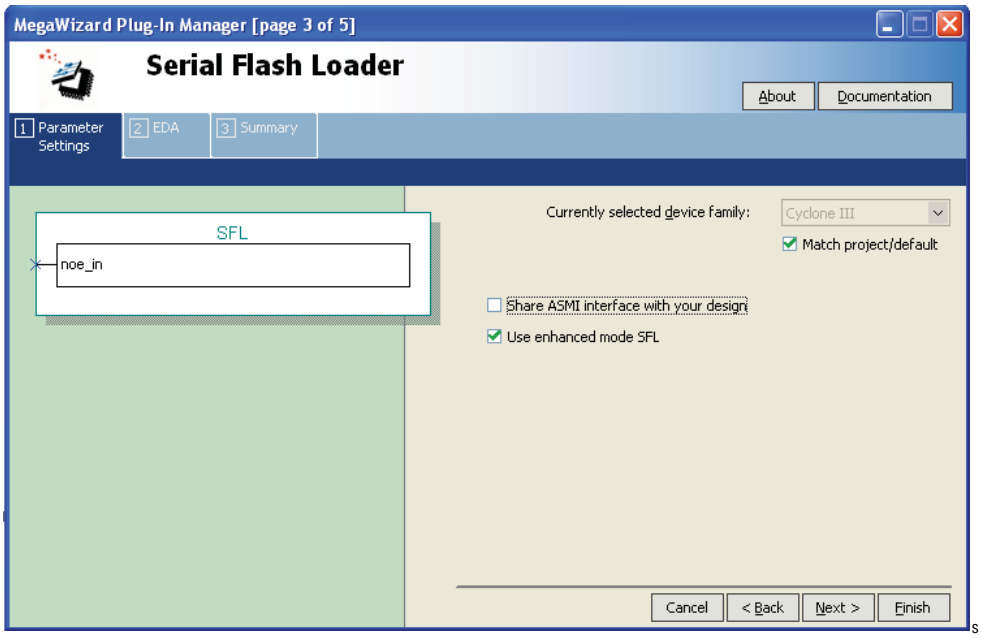
1. On the Tools menu, click **MegaWizard Plug-In Manager**. Page 1 of MegaWizard Plug-In Manager appears**.**
2. Select the **Create a new custom megafunction variation** option and click **Next**. Page 2a of the MegaWizard Plug-In Manager appears, as shown in Figure 5.
3. Select the FPGA device family from the **Which device family will you be using?** pull-down list.
4. Select **Serial Flash Loader** from the **JTAG-accessible Extensions** category in the megafunction list.
5. Select the Hardware Description Language (HDL) output file type and name the file. Click **Next** (Verilog HDL was chosen for this example).

Figure 5. SFL Megafunction Settings


MegaWizard Plug-In Manager [page 2a]
Which megafunction would you like to customize?
Select a megafunction from the list below
Installed Plug-Ins
Advanced SEU Detection
Altera SOPC Builder
Arithmetic
Communications
DSP
Gates
I/O
Interfaces
JTAG-accessible Extensions
In-System Sources and Probes
Parallel Flash Loader
Serial Flash Loader
SignalTap II Logic Analyzer
Virtual JTAG
Memory Compiler
Storage
IP MegaStore
Which device family will you be using?
Stratix III
Which type of output file do you want to create?
AHDL
VHDL
Verilog HDL
What name do you want for the output file?
Browse...
D:\My Documents\Project\SFL
Return to this page for another create operation
Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in the global user libraries specified in the Options dialog box (Tools menu), or a user library specified in the User Libraries page of the Settings dialog box (Assignments menu).
Your current user library directories are:
Project User Libraries:
c:\altera\71\ip\ddr_high_perf\lib\
sources\clk_ctrl\clk_ctrl_pll\
sources\clk_ctrl\clk_ctrl_out_altclkctrl\
sources\clk_ctrl\clk_ctrl_altclkctrl\
sources\clk_ctrl\
sources\
Cancel
< Back
Next >
Finish

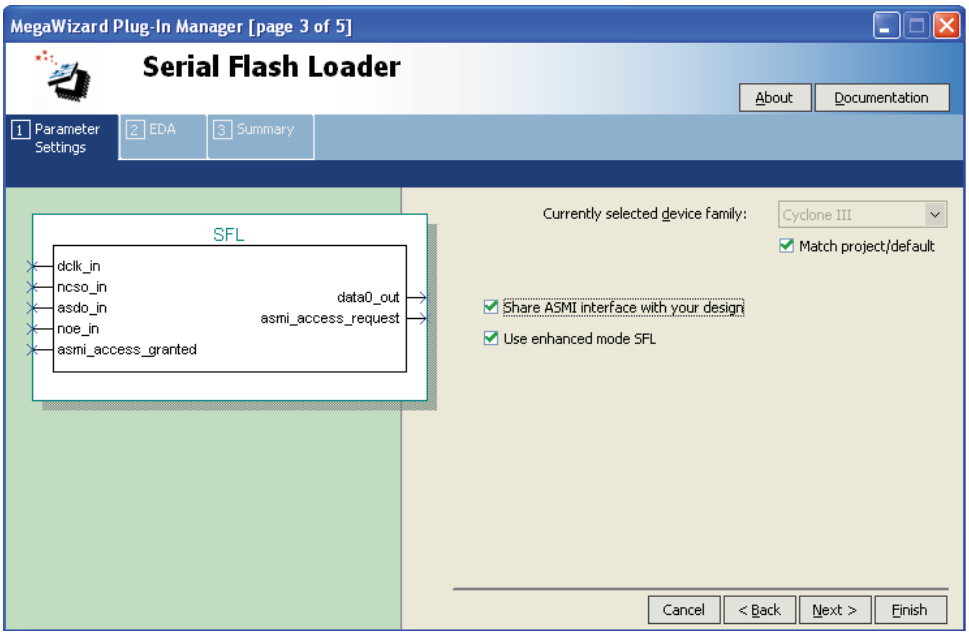
6. Specify the directory and output filename. Click **Next**. Page 3 of the MegaWizard Plug-In Manager appears, as shown in Figure 6.

Figure 6. SFL Megafunction Parameter Settings



7. Turn on the **Share ASMI interface in the design** check box if you must share the ASMI interface with your design. This option provides additional control pins for controlling the ASMI interface, as shown in Figure 7.
8. The **Use enhanced mode SFL** check box is turned on by default. This option provides more flexibility for JTAG cascading environment and the usage of the SFL with a third-party programmer tool. Turn off the **Use enhanced mode SFL** check box if you do not wish to use enhanced SFL.

Figure 7. SFL Megafunction with the "Share ASMI interface with your design" Option



9. Click **Next** until you reach the summary page.
10. Click **Finish** to generate the SFL megafunction. The Quartus II software generates the megafunction in the form of the HDL file you specified.

The SFL megafunction does not have any timing or simulation model. Therefore, it cannot be simulated.

Table 2 shows the input and output signals for the SFL megafunction.

Table 2. Input and Output Signals for the SFL Megafunction (Part 1 of 2)

Signal	Input/Output	Description
dclk_in *(1)*	Input	Clock signal from user design to DCLK.
ncso_in *(1)*	Input	Control signal from user design to nCSO pin. A low signal enables the EPCS.
asdo_in *(1)*	Input	Control signal from user design to ASDO pin for sending data into EPCS.

Table 2. Input and Output Signals for the SFL Megafunction (Part 2 of 2)

Signal	Input/Output	Description
noe_in	Input	Control signal to enable the SFL Megafunction. A low signal enables the megafunction. SFL tri-states ASMI interface when it is disabled. You can leave this signal tied to GND all the time if you do not wish to gain access to ASMI interface.
asmi_access_granted *(1)*	Input	Control signal to allow SFL to access the DCLK, nCSO, ADSO and DATA0 pins using the ASMI interface. A high signal allows SFL to access the ASMI interface. A low signal allows user design to access to ASMI interface.
data0_out *(1)*	Output	Signal from DATA0 pin to user design.
asmi_access_request *(1)*	Output	A high signal indicates SFL is requesting ASMI interface access. SFL starts accessing ASMI interface when ASMI_ACCESS_GRANTED is high.

Note for Table 2:

(1) These ports are available when the **Share ASMI interface with your design** check box is turned on in the megafunction.

Generating .jic and .jam Programming Files in the Quartus II Software

You can program serial configuration devices with either **.jic**, **.jam**, or **.jbc** programming files using the Quartus II programmer. To generate **.jic** or **.jam** programming files with the Quartus II software, you must first generate a user-specified SRAM object file (**.sof**), which is the input file. Next, you must convert the **.sof** to a **.jic** file. Alternatively, if you prefer to use **.jam** programming files, you must convert the **.jic** file to a **.jam** file. This section provides the following instructions:

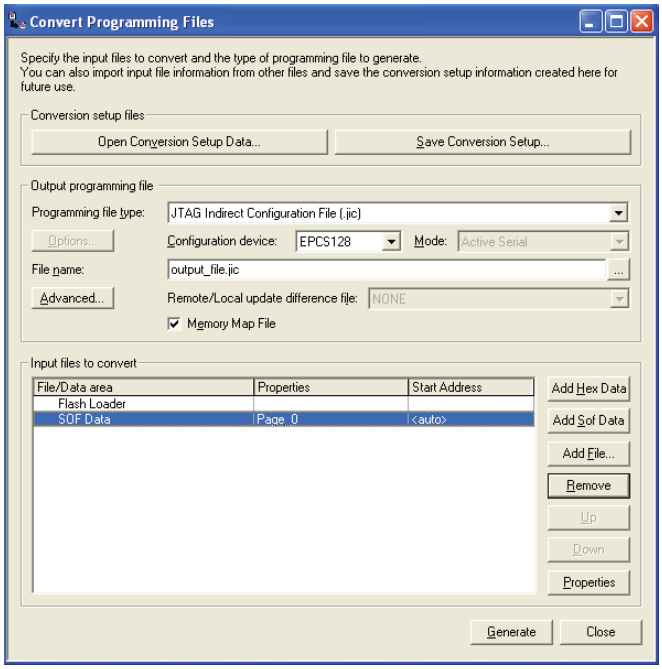
- "Converting .sof to .jic Files in the Quartus II Software" on page 10
- "Converting .jic Files to .jam Files in the Quartus II Software" on page 13

Converting .sof to .jic Files in the Quartus II Software

To convert a **.sof** to a **.jic** file, perform the following steps:

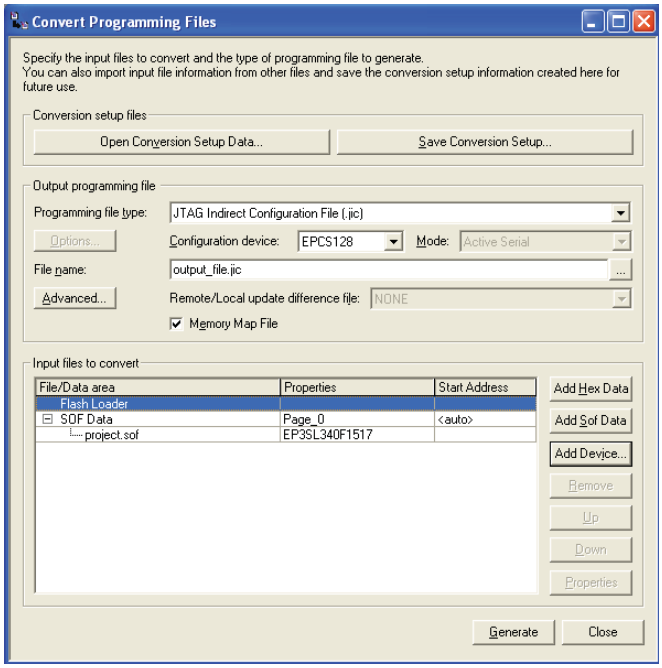
1. On the File menu, select **Convert Programming Files**.
2. In the **Convert Programming Files** dialog box, select **JTAG Indirect Configuration File (.jic)** from the **Programming file type** drop down menu.
3. In the **Configuration device** field, specify the targeted serial configuration device.
4. In the **File name** field, browse to the target directory and specify an output file name.
5. Highlight the **SOF Data** in the Input files to convert window (refer to Figure 8).

Figure 8. Convert Programming Files Dialog Box



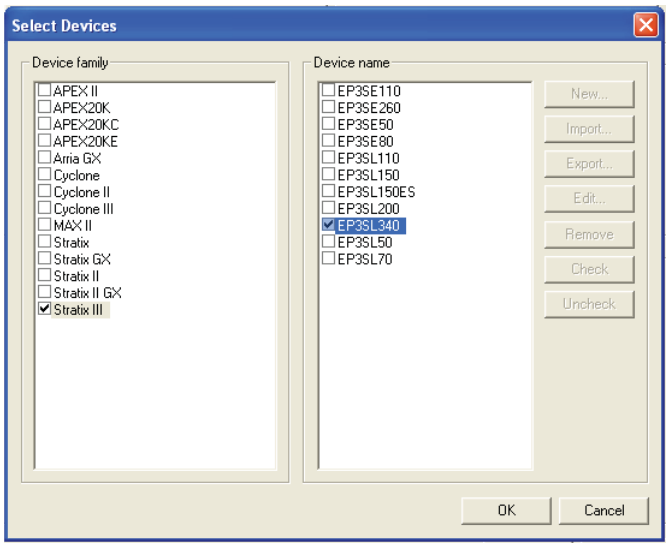
6. Click **Add File**.
7. Select the **.sof** file that you want to convert to a **.jic** file.
8. Click **OK**.
9. Highlight **FlashLoader** and click **Add Device**, as shown in Figure 9.

Figure 9. Highlight FlashLoader



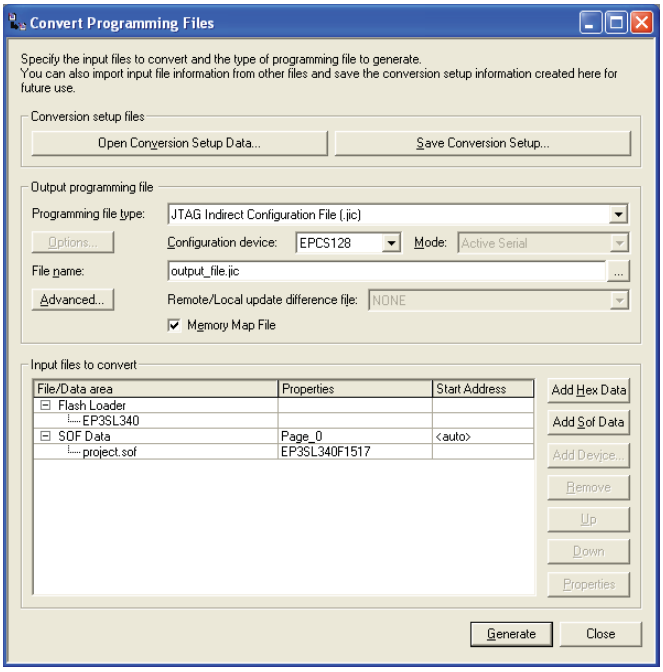
10. Click **OK**. The **Select Devices** dialog box appears.
11. Select the targeted FPGA that you are using to program the serial configuration device (refer to Figure 10).

Figure 10. Select Devices Dialog Box



12. Click **OK**. The **Convert Programming Files** dialog box appears (refer to Figure 11).

Figure 11. Convert Programming Files Dialog Box



13. Click **Generate**.

The **Memory Map File** check box is checked by default. The Quartus II programmer generates the memory allocation mapping file along with the **.jic** file. You can turn off this option by turning off the check box.

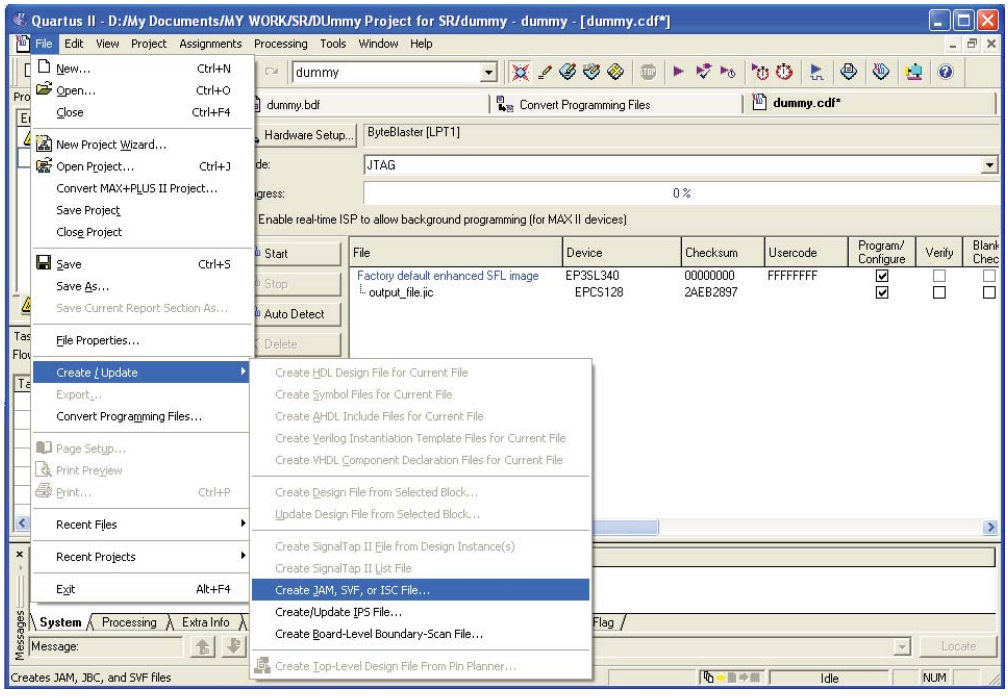
To program the serial configuration device or devices with the **.jic** file that you created, add the file to the Quartus II programmer window and perform the steps in "Programming Serial Configuration Devices with the Quartus II Programmer" on page 14.

Converting .jic Files to .jam Files in the Quartus II Software

To convert a **.jic** to a **.jam** file in the Quartus II software, perform the following steps:

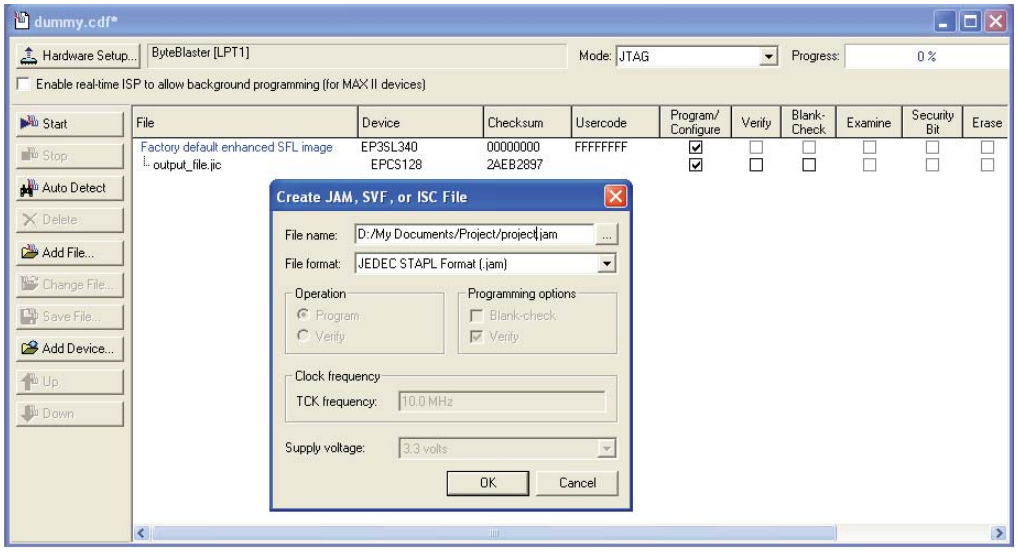
1. On the Tools menu, select **Programmer**.
2. Click **Add File**. The **Select Programming File** dialog box appears.
3. Browse to the **.jic** file you created in "Converting .sof to .jic Files in the Quartus II Software" on page 10. Add more **.jic** files if you are programming multiple serial configuration devices.
4. Click **Open**.
5. Select **Create/Update**. In the File menu, scroll to **Create JAM, SVF, or ISC File** (refer to Figure 12).

Figure 12. Create JAM, SVF, or ISC File



6. The **Create JAM, SVF, or ISC File** dialog box appears, as shown in Figure 13.

Figure 13. Converting a .jic File to a .jam File in the Quartus II Software



7. Click **OK**.

To program the serial configuration device or devices with the **.jam** file that you created, add the file to the Quartus II programmer window and perform the steps in "Programming Serial Configuration Devices Using the Quartus II Programmer and .jam Files" on page 18.

With the same steps outlined above, you can generate a **.jbc** or **.svf** file from the **.jic** file.

Programming Serial Configuration Devices with the Quartus II Programmer

You can use the Quartus II programmer to generate serial configuration device programming files. The Quartus II programmer can generate both **.jic** and **.jam** files with the factory default enhanced SFL image that is run directly from the Quartus II programmer.

As long as the JTAG interface of the FPGA is accessible for programming, you can use the factory default enhanced SFL image that is run directly from the Quartus II programmer for your application. If you are using the design security feature with the tamper-protection bit set into which the JTAG interface is inaccessible for programming, the factory default enhanced SFL image will not work.

This section discusses the following topics:

- "Programming Serial Configuration Devices Using the Quartus II Programmer and .jic Files" on page 15
- "Programming Serial Configuration Devices Using the Quartus II Programmer and .jam Files" on page 18

Programming Serial Configuration Devices Using the Quartus II Programmer and .jic Files

To program serial configuration devices with **.jic** files, you must perform the following steps:

1. When the **.sof**-to-**.jic** file conversion is complete (refer to Figure 11 on page 12), add the **.jic** file to the Quartus II programmer window:
   a. In the Tools menu, choose **Programmer**. The **Chain1.cdf** dialog box appears.
   b. Click **Add File**. In the **Select Programming File** dialog box, browse to the **.jic** file.
   c. Click **Open**.
2. Configure the FPGA with the SFL image by turning on the FPGA **Program/Configure** box (refer to Figure 14). This process corresponds to Step 1 of Figure 3 on page 5. After the **Program/Configure** box is turned on, the Quartus II programmer automatically invokes the factory default enhanced SFL image.

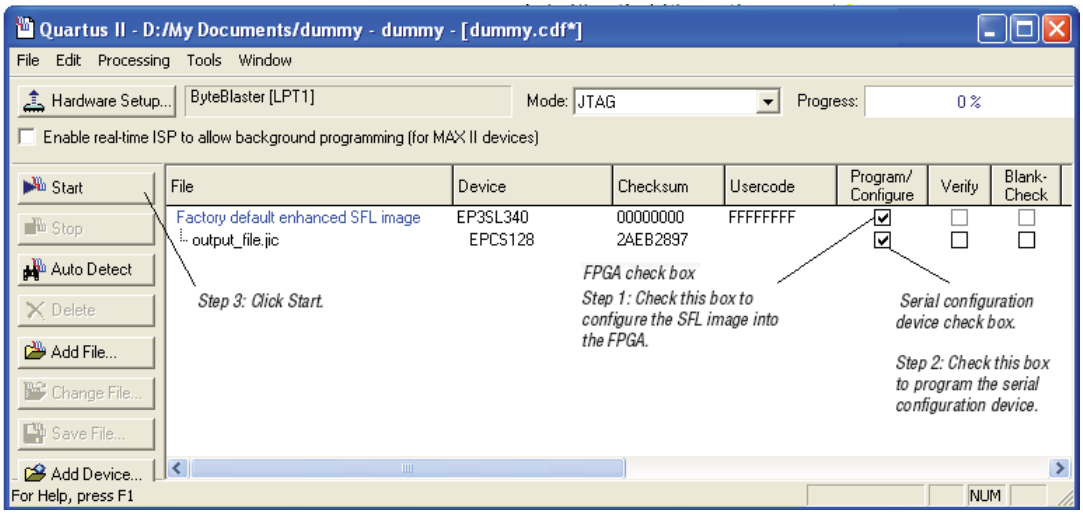
   ☞ By default, the factory default enhanced SFL image invokes directly from the Quartus II programmer after the **Program/Configure** check box is turned on. To revert back to legacy SFL (Factory Default SFL image), turn off the **Use enhanced Serial Flash Loader (SFL) IP as factory default image** check box under **Tools** --> **Options** --> **Programmer** tab, as shown in Figure 16.

   ☞ The **Check block CRCs to accelerate PFL/SFL verification when available** check box under **Tools** --> **Options** --> **Programmer** tab (Figure 16) is another relevant option for the enhanced mode SFL solution. This check box is turned on by default to speed up the EPCS image verification process using CRC method. The verification takes place when you turn on the **Verify** check box in the Quartus II programmer. If you do not wish to use this option, turn off the check box.

3. Program the serial configuration device by turning on the corresponding **Program/Configure** box (refer to Figure 14). This process corresponds to Step 2 of Figure 3 on page 5.
4. Click **Start**.

Figure 14 shows the Quartus II programmer window with one **.jic** file.

Figure 14. Quartus II Programmer Window with One .jic File



If the **Program/Configure** check boxes are not specified, the Quartus II programmer bypasses the request. Also, if the FPGA does not have the SFL image when the serial configuration device data is programmed via the JTAG interface, the programming process will fail.

You can program multiple serial configuration devices by including more than one **.jic** file in the Quartus II programmer.

FPGA must be in active serial configuration mode to enable SFL to program.

Figure 15 shows the Quartus II programmer window with multiple **.jic** files.

Figure 15. Quartus II Programmer Window with Multiple .jic Files

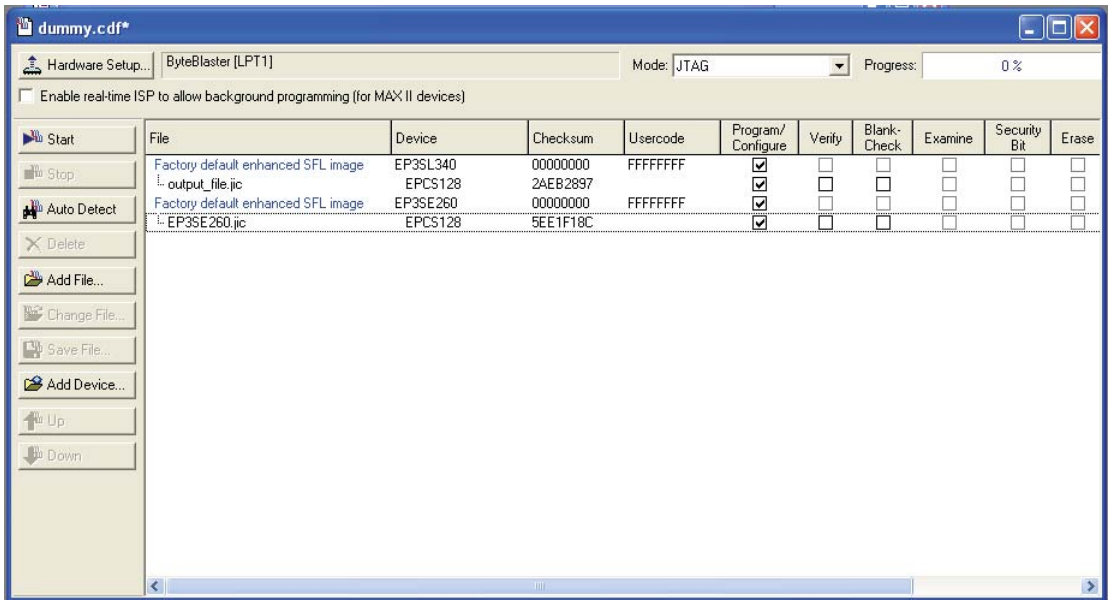
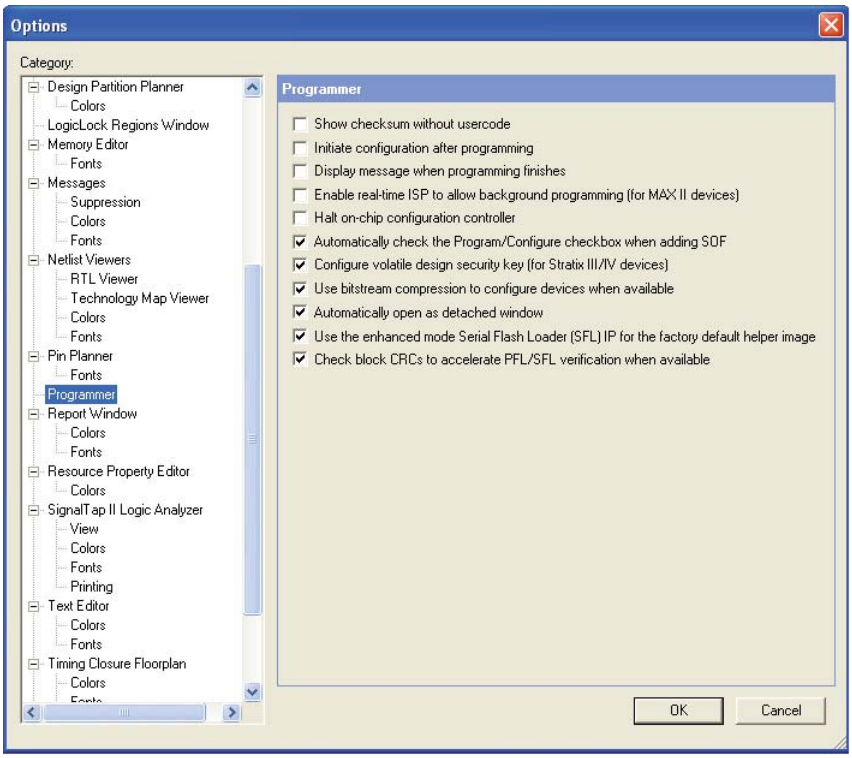


Figure 16 shows the Quartus II programmer options default settings.

Figure 16. Quartus II Programmer Options Default Settings



Programming Serial Configuration Devices Using the Quartus II Programmer and .jam Files

When programming with **.jam** files, the Quartus II programmer requires configuring the FPGA and programming the serial configuration device in one step. Therefore, Figure 17 shows just one **Program/Configure** check box.

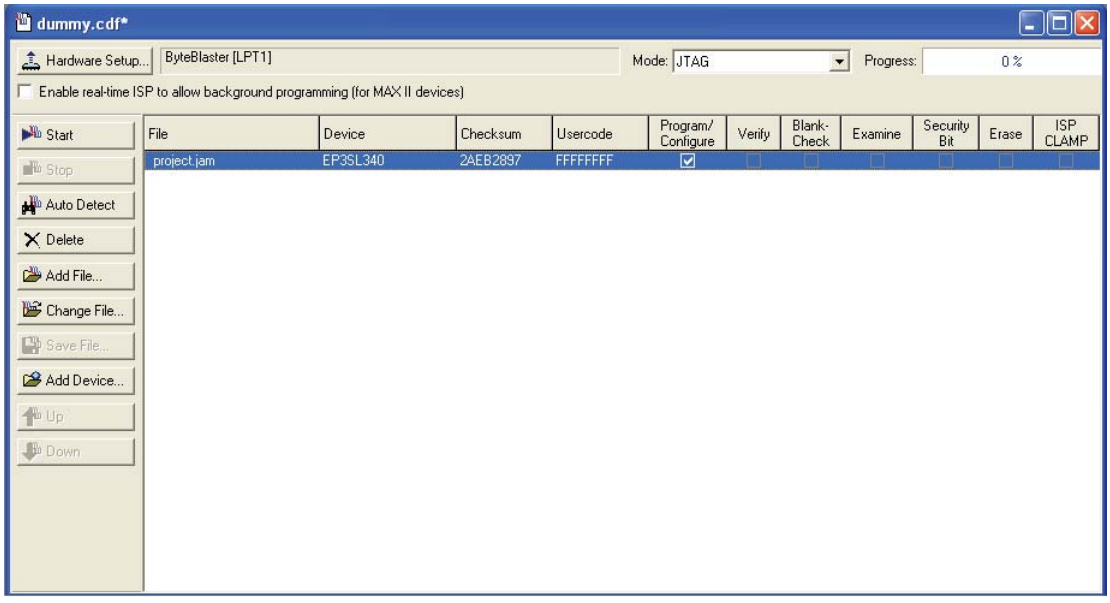
To program serial configuration devices with **.jam** file, perform the following steps:

1. When the **.jic**-to-**.jam** file conversion is complete (refer to Figure 13 on page 14), add the **.jam** file to the Quartus II programmer window:
   a. In the Tools menu, select **Programmer**. The **Chain1.cdf** dialog box appears.
   b. Click **Add File**. In the **Select Programming File** dialog box, browse to the **.jam** file.
   c. Click **Open**.
2. Configure the FPGA with the SFL image, and program the serial configuration device by turning on the FPGA **Program/Configure** check box (refer to Figure 17). This process corresponds to Step 1 and Step 2 of Figure 3 on page 5.
3. Click **Start**.

The **.jam** file is generated from the **.jic** file via the chain description file (**.cdf**). For more information, refer to Quartus II Help.

Figure 17 shows the Quartus II programmer window with one **.jam** file.

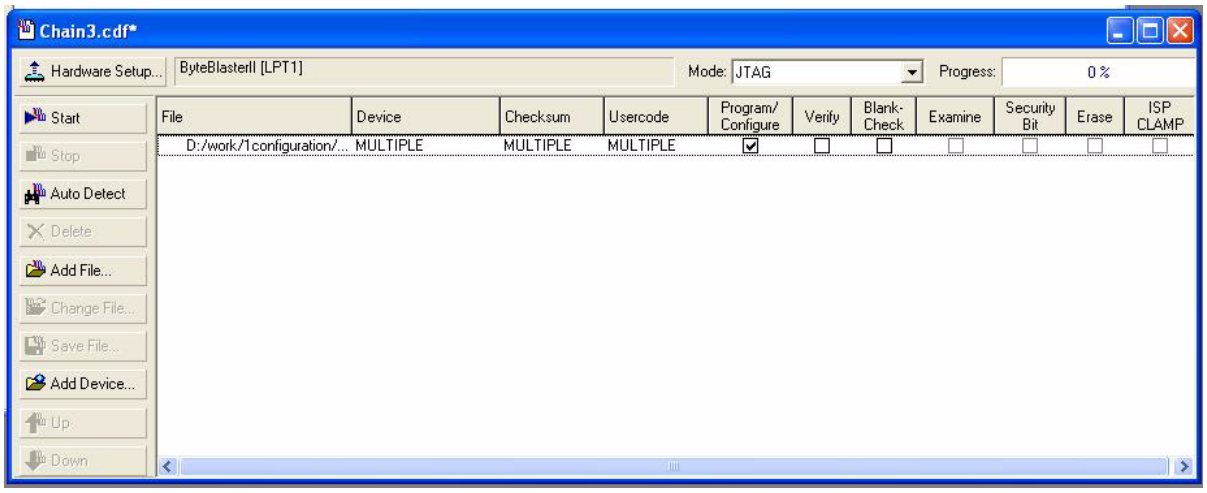
Figure 17. Quartus II Programmer Window with One .jam File



You can program multiple serial configuration devices with one **.jam** file in the Quartus II programmer.

Figure 18 shows the Quartus II programmer window with one **.jam** file programming multiple serial configuration devices.

Figure 18. Quartus II Programmer Window with One .jam File Programming Multiple Serial Configuration Devices



Conclusion

The SFL offers an in-system programming solution for serial configuration devices via the JTAG interface. Because the JTAG is an industry standard interface, it is preferred over the AS interface. Thus, in addition to the conventional method of programming serial configuration devices via the AS interface, you now have the option to use the JTAG interface and the SFL solution.

Document Revision History

Table 3 shows the revision history for this application note.

Table 3. Document Revision History (Part 1 of 2)

Date and Revision	Changes Made	Summary of Changes
April 2009 version 3.1	■ Updated the "Introduction" section ■ Updated the "Using the SFL Megafunction in the Quartus II Software" section ■ Updated Figure 5 and Figure 7 in the "Instantiating SFL Megafunction in the Quartus II Software" ■ Updated Figure 8, Figure 9, Figure 10, and Figure 11 in the "Converting .sof to .jic Files in the Quartus II Software" section ■ Updated Figure 12 and Figure 13 in the "Converting .jic Files to .jam Files in the Quartus II Software" section ■ Updated Figure 14 and Figure 15 in the "Programming Serial Configuration Devices Using the Quartus II Programmer and .jic Files" section ■ Updated Figure 17 in the"Programming Serial Configuration Devices Using the Quartus II Programmer and .jam Files" section ■ Updated the "Instantiating SFL Megafunction in the Quartus II Software" section ■ Updated Figure 6 in "Instantiating SFL Megafunction in the Quartus II Software" section ■ Updated Table 2 in the "Instantiating SFL Megafunction in the Quartus II Software" section ■ Added handnote to the "Converting .sof to .jic Files in the Quartus II Software" section ■ Added handnote to the "Converting .jic Files to .jam Files in the Quartus II Software" section ■ Updated the "Programming Serial Configuration Devices with the Quartus II Programmer" section ■ Updated the "Programming Serial Configuration Devices Using the Quartus II Programmer and .jic Files" section ■ Updated the "Programming Single and Multiple Serial Configuration Devices with the SFL Solution" section	—

Table 3. Document Revision History (Part 2 of 2)

Date and Revision	Changes Made	Summary of Changes
July 2006 version 3.0	■ Updated the first paragraph in the "Introduction" section ■ Updated the first column of Table 1 ■ Updated the forth paragraph in the "Introduction" section ■ Updated the bulleted list in the "Introduction" section ■ Added Note to Step 2 of the "Steps for Programming Single and Multiple Serial Configuration Devices with the SFL Solution"section ■ Added Figure 2 to the"Steps for Programming Single and Multiple Serial Configuration Devices with the SFL Solution" section ■ Added notes to Figure 3 and Figure 4 ■ Added the "Using the SFL Megafunction in the Quartus II Software" section ■ Added a note to Figure 11 and after Figure 11	Changes in response to first user feedback
June 2008, version 2.0	■ Updated the first and forth paragraph and the bulleted list in the "Introduction"section ■ Updated column one of Table 1 ■ Updated steps 2 and 3 in the "Steps for Programming Single and Multiple Serial Configuration Devices with the SFL Solution" section	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Copyright © 2009 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

I.S. EN ISO 9001